



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

| APPLICATION NO. | FILING DATE | FIRST NAMED INVENTOR | ATTORNEY DOCKET NO. | CONFIRMATION NO. |
|-----------------|-------------|----------------------|---------------------|------------------|
|-----------------|-------------|----------------------|---------------------|------------------|

10/797,515

03/10/2004

Steve Naroff

772.025US1

4252

21186

7590

11/01/2006

SCHWEGMAN, LUNDBERG, WOESSNER & KLUTH, P.A.
P.O. BOX 2938
MINNEAPOLIS, MN 55402

EXAMINER

NGUYEN, PHILLIP H .

ART UNIT

PAPER NUMBER

2191

DATE MAILED: 11/01/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary

Application No.

10/797,515

Applicant(s)

NAROFF ET AL.

Examiner

Phillip H. Nguyen

Art Unit

2193

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 10 March 2004.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-51 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-51 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☒ The specification is objected to by the Examiner.
- 10) ☒ The drawing(s) filed on 10 March 2004 is/are: a) ☐ accepted or b) ☒ objected to by the Examiner.
- Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
- Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
 2. ☐ Certified copies of the priority documents have been received in Application No. _____.
 3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- | | |
|--|---|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892) | 4) <input type="checkbox"/> Interview Summary (PTO-413) |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | Paper No(s)/Mail Date. _____ |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08) | 5) <input type="checkbox"/> Notice of Informal Patent Application |
| Paper No(s)/Mail Date _____ | 6) <input type="checkbox"/> Other: _____ |

DETAILED ACTION

1. This action is in response to the original filing of March 10, 2004. Claims 1-51 are pending and have been considered below.

Drawings

2. The drawings are objected to as failing to comply with 37 CFR 1.84(p)(5) because they do not include the following reference sign(s) mentioned in the description: Figure 1, item 112. Corrected drawing sheets in compliance with 37 CFR 1.121(d) are required in reply to the Office action to avoid abandonment of the application. Any amended replacement drawing sheet should include all of the figures appearing on the immediate prior version of the sheet, even if only one figure is being amended. Each drawing sheet submitted after the filing date of an application must be labeled in the top margin as either "Replacement Sheet" or "New Sheet" pursuant to 37 CFR 1.121(d). If the changes are not accepted by the examiner, the applicant will be notified and informed of any required corrective action in the next Office action. The objection to the drawings will not be held in abeyance.

3. The drawings are objected to as failing to comply with 37 CFR 1.84(p)(5) because they include the following reference character(s) not mentioned in the description: Figure 10, item 108. Corrected drawing sheets in compliance with 37 CFR 1.121(d), or amendment to the specification to add the reference character(s) in the description in compliance with 37 CFR 1.121(b) are required in reply to the Office action

Art Unit: 2193

to avoid abandonment of the application. Any amended replacement drawing sheet should include all of the figures appearing on the immediate prior version of the sheet, even if only one figure is being amended. Each drawing sheet submitted after the filing date of an application must be labeled in the top margin as either "Replacement Sheet" or "New Sheet" pursuant to 37 CFR 1.121(d). If the changes are not accepted by the examiner, the applicant will be notified and informed of any required corrective action in the next Office action. The objection to the drawings will not be held in abeyance.

Specification

4. The disclosure is objected to because of the following informalities:

In page 6, paragraph 4; the applicant recites a loader reference 112 corresponding to figure 1, but item 112 does not exist in figure 1. The examiner suggests the applicant to add item 112 to figure 1.

In page 10, paragraph 3; the applicant recites, "Based on the source code module, the compiler unit 204 produces an executable object module (e.g., executable object module 212A)." The examiner suggests the applicant takes out "executable" and change 212A to 214A.

In page 16, paragraph 2; the applicant recites item 310 responding to figure 6 in the drawings, but item 310 does not exist in figure 6. The applicant is required to correct the specification to corresponding to figure 6.

In page 20, paragraph 4; the applicant recites item 1008 corresponding to figure 10 in the drawings, but item 1008 does not exist in figure 10. The applicant is required to either correct the drawings or the specification to corresponding to each other.

Appropriate correction is required.

Claim Rejections - 35 USC § 101

5. 35 U.S.C. 101 reads as follows:

Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

Claims 30-45 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter. These claims are non-statutory because they are directed to a machine-readable medium, which is disclosed as propagated signals (e.g., carrier waves, infrared signals, digital signals, etc.). The specification provides intrinsic evidence the machine-readable medium is intended to cover propagated signals (e.g., carrier waves, infrared signals, digital signals, etc. in page 9, paragraph 2), such are currently not believed to enable the machine-readable medium to act as a computer component and realize its functionality absent being claimed in combination with the necessary hardware to receive and convert the propagated signals to computer usable code.

Claim Rejections - 35 USC § 112

6. The following is a quotation of the second paragraph of 35 U.S.C. 112:

The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

7. Claim 21 is rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

The applicant recites the limitation "wherein the loader unit is to use the address to load the second set of instructions into the memory unit." It is unclear to the examiner whether the applicant refers to the address of the third set of instructions or the address of the second set of instructions. For the examining purposes, the examiner interprets the address as the address of the second set of instructions.

Claim Rejections - 35 USC § 102

8. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

9. Claims 1, 2, 4-8, 17, 21, 22, 24-26, 30, 31, 33-36, 39, 46, 47, and 49-51 are rejected under 35 U.S.C. 102(b) as being anticipated by Stoodley (US 6,813,764).

Claim 1: Stoodley discloses a method comprising:

a. although, Stoodley does not explicitly disclose loading a first set of instructions into an execution unit. It is inherent in Stoodley's computer system in order to execute the compiled code emitted by the compiler and replace the unresolved reference; wherein the first set of instructions includes an unresolved reference to a second set of instructions ("compiled code including a set of target instructions...each target instruction including an unresolved reference", Col 2, line 18-21, Stoodley does not explicitly disclose unresolved reference to a second set of instructions. It is inherent in Stoodley's approach in order to fulfill the purpose of his invention), wherein the loading includes replacing the unresolved reference with an address of a third set of instructions ("replacing the unresolved code in the compiled code with a call instruction for calling the snippet", Col 3, line 54-55, snippet is a third set of instruction);

b. executing instructions of the first set ("on execution..." Col 2, line 28);

c. executing instructions of the third set ("at runtime" Col 3, line 28-30); where executing instructions of the third set includes loading the second set of instructions into the execution unit ("snippet including instructions and data to identify the unresolved code in the compiled code and to call and provide data to the runtime helper" Col 3, line 48-50, runtime helper is a second set of instructions, gets loaded into the execution unit when snippet calls it); and

d. executing instructions of the second set ("runtime helper" Col 2, line 15, snippet calls/loads the runtime helper into the execution unit to resolve unresolved references . . .

Claim 2: Stoodley discloses the method as in claim 1 above, and further discloses the first set of instructions includes an executable object module ("compiled code" Col 2, line 18).

Claim 4: Stoodley discloses the method as in claim 1 above, and further discloses the second set of instructions includes a separately compiled object module ("an associated runtime helper being defined for resolving references at runtime" Col 2, line 1-2, the runtime helper is a second module and separately from the compiled code).

Claim 5: Stoodley discloses the method as in claim 1 above, but does not explicitly disclose the third set of instructions includes a loader unit. It is inherent in Stoodley's computer system ("snippet includes instructions to pass the resolution data to the runtime helper" Col 2, line 54-55; when the snippet passes resolution data to the helper code, it's actually performing the loading data to the helper code).

Claim 6: although, Stoodley discloses the method as in claim 1 above, but does not explicitly disclose the loading does not include determining whether the unresolved reference refers to a defined external symbol. It is inherent in Stoodley's approach because his computer system identifies the unresolved reference and resolves them at runtime ("identify the unresolved reference and to permit the reference to be resolved by instructions in runtime helper" Col 5, line 5-7).

Claim 7: Stoodley discloses a method comprising:

a. compiling a source code module into an executable object module ("compiled code" Col 2, line 16-17) that includes an unresolved reference to a separately compiled object module ("each target instruction including an unresolved reference" Col 2, line 19-20, Stoodley does not explicitly disclose unresolved reference to a second set of instructions." It is inherent in Stoodley's approach in order to fulfill the purpose of his invention);

b. although, Stoodley does not explicitly disclose loading the executable object module. It is inherent in Stoodley's computer system in order to execute the compiled code and replace the unresolved reference; wherein the loading includes replacing the unresolved reference with a reference to a system module ("replacing the unresolved code in the compiled code with a call instruction for call the snippet", Col 3, line 54-55, snippet is a system module), and wherein neither the compiling nor the loading including determining whether the unresolved reference refers to a defined external symbol.

c. executing the executable object module ("on execution" Col 2, line 28); wherein the executing includes, calling the system module for loading the separately compiled object module ("call to the corresponding snippet and in which the snippet includes instructions to pass the resolution data to helper code", Col 2, line 53-55); and

d. executing the separately compiled object module ("runtime helper" Col 2, line 15, snippet calls/loads the runtime helper into the execution unit to resolve unresolved reference).

Claim 8: Stoodley discloses the system as in claim 7 above, but does not explicitly disclose the system module includes a loader unit. It is inherent in Stoodley ("snippet includes instructions to pass the resolution data to the runtime helper" Col 2, line 54-55; when the snippet passes resolution data to the helper code, it's actually performing the loading data to the helper code).

Claim 17: Stoodley discloses an apparatus comprising:

a. a compiler unit to create an executable object module based on a source code module ("the compiler including a means to emit compiled code" Col 2, line 16-17), wherein the executable object module includes an unresolved reference to a separately compiled object module ("each target instruction including an unresolved reference" Col 2, line 19-20);

b. although, Stoodley does not explicitly disclose a storage unit to store the executable object module. It is inherent in Stoodley's computer system in order to store the executable object module after the compiler emits them.

c. although, Stoodley does not explicitly disclose an execution unit to receive the executable object module. It is inherent in Stoodley's computer system ("on execution" Col 2, line 28) in order to execute the executable object module ("compiled code"); and

d. although, Stoodley does not explicitly disclose a loader unit to find the executable object module in the storage unit and present the executable object module to the execution unit. It is inherent in Stoodley's computer system because in order to execute the executable object module the loader unit must load the compiled code from

the storage to execution unit; wherein the loader unit is to replace the unresolved reference with a reference to a system module ("replacing the unresolved code in the compiled code with a call instruction for call the snippet" Col 3, line 53-54, snippet is a system module), and wherein the loader unit is not a determine whether the unresolved reference refers to a defined external object module.

Claim 21: Stoodley disclose an apparatus comprising:

- a. although, Stoodley does not explicitly disclose a loader unit to load a first set of instructions into a memory unit. It is inherent in Stoodley's computer system because the compiled code must be loaded into a memory unit after emitted by the compiler; wherein the first set of instructions includes an unresolved reference to a second set of instructions ("each target instruction including an unresolved reference" Col 2, line 20-21), the loader unit to replace the unresolved reference with an address of a third set of instructions ("replacing the unresolved code in the compiled code with a call instruction for calling the snippet" Col 3, line 54-55, snippet is a third set of instructions); and
- b. an execution unit to execute instructions of the first set ("on execution" Col 2, line 28); the execution unit also to execute instructions of the third set to determined an address of the second set of instructions ("snippet including instructions and data to identify the unresolved reference" Col 3, line 48-49); wherein the loader unit is to use the address to load the second set of instructions into the memory unit ("call and provide data to the runtime helper" Col 3, line 50, the runtime helper is a second module, gets loaded into the memory unit for executing when snippet module calls it).

Claim 22: Stoodley discloses the apparatus as in claim 21 above, and further discloses the first set of instructions is an executable object module ("compiled code" Col 2, line 18).

Claim 24: Stoodley discloses the apparatus as in claim 21 above, and further discloses the second set of instructions is a separately compiled object module ("an associated runtime helper being defined for resolving references at runtime" Col 2, line 1-2, the runtime helper is a separated from the compiled code emitted by the compiler).

Claim 25: Stoodley discloses the apparatus as in claim 21 above, but does not explicitly disclose the third set of instructions is a loader unit. It is inherent in Stoodley's computer system ("snippet includes instructions to pass the resolution data to the runtime helper" Col 2, line 54-55; when the snippet passes resolution data to the helper code, it's actually performing the loading data to the helper code).

Claim 26: Stoodley discloses a system comprising:

a. although, Stoodley does not explicitly disclose a memory unit. It is inherent in Stoodley's computer system in order to store the compiled code emitted by the compiler, the memory unit including a compiler unit ("a compiler" Col 2, line 14) to create an executable object module based on a source code module ("compiled code" Col 2, line 17); wherein the executable object module includes a symbolic reference to a

separately compiled object module ("compiled code including a set of instructions... each target instruction including an unresolved reference" Col 2, line 17-20); and

b. although, Stoodley does not explicitly disclose a loader unit to present the executable object module for execution. It is inherent in Stoodley's computer system in order to present the executable object module to the execution unit; wherein the loader unit is to replace the symbolic reference with an address to a system module ("replacing the unresolved code in the compiled code with a call instruction for calling the snippet" Col 3, line 54-55); and wherein the loader unit is not to determine whether the symbolic reference refers to a defined external object module; and

c. although, Stoodley does not explicitly disclose "a processor to receive the executable object module from the loader unit of the memory unit." It is inherent in Stoodley's computer system in order to process the compiled code from the loader.

Claim 30: Stoodley discloses a machine-readable medium that provides instructions, which when executed by a machine, cause the machine to perform operations comprising:

a. although, Stoodley does not explicitly disclose loading a first set of instructions into an execution unit. It is inherent in Stoodley's computer system in order to execute the compiled code emitted by the compiler and replace the unresolved reference; wherein the first set of instructions includes an unresolved reference to a second set of instructions ("compiled code including a set of target instructions...each target instruction including an unresolved reference", Col 2, line 18-21, Stoodley does

Art Unit: 2193

not explicitly disclose an unresolved reference to a second set of instructions. It is inherent in Stoodley's approach in order to fulfill the purpose of his invention), wherein the loading includes replacing the unresolved reference with an address of a third set of instructions ("replacing the unresolved code in the compiled code with a call instruction for calling the snippet", Col 3, line 54-55, snippet is a third set of instruction);

b. executing instructions of the first set ("on execution", Col 2, line 28);

c. executing instructions of the third set ("at runtime", Col 3, line 28-30); where executing instructions of the third set includes loading the second set of instructions into the execution unit ("snippet including instructions and data to identify the unresolved reference in the compiled code and to call and provide data to the runtime helper", Col 3, line 48-50, runtime helper is a second module and gets loaded into the execution unit when snippet calls it); and

d. executing instructions of the second set ("runtime helper" Col 2, line 15, snippet calls/loads the runtime helper into the execution unit to resolve unresolved reference.

Claim 31: Stoodley discloses the machine-readable medium as in claim 30 above, and further discloses the first set of instructions includes an executable object module ("compiled code" Col 2, line 18).

Claim 33: although, Stoodley discloses the machine-readable medium as in claim 30 above, but does not explicitly disclose the loading does not include determining

Art Unit: 2193

whether the unresolved reference refers to a defined external symbol. It is inherent in Stoodley's approach because his computer system identifies the unresolved reference and resolves them at runtime ("identify the unresolved reference and to permit the reference to be resolved by instructions in runtime helper" Col 5, line 5-7).

Claim 34: Stoodley discloses the machine-readable medium as in claim 30 above, and further discloses the second set of instructions includes a separately compiled object module ("an associated runtime helper being defined for resolving references at runtime" Col 2, line 1-2, the runtime helper is a second module and separately from the compiled code).

Claim 35: Stoodley discloses the machine-readable medium as in claim 30 above, but does not explicitly disclose the third set of instructions includes a loader unit. It is inherent in Stoodley's computer system ("snippet includes instructions to pass the resolution data to the runtime helper" Col 2, line 54-55; when the snippet passes resolution data to the helper code, it's actually performing the loading data to the helper code).

Claim 36: Stoodley discloses a machine-readable medium that provides instructions, which when executed by a machine, cause the machine to perform operations comprising:

a. compiling a source code module into an executable object module ("compiled code" Col 2, line 16-17) that includes an unresolved reference to a separately compiled object module ("each target instruction including an unresolved reference" Col 2, line 19-20) Stoodley does not explicitly disclose an unresolved reference to a separately compiled object module. It is inherent in Stoodley's approach because an external function call or reference to external variable is call to a separately compiled object module);

b. although, Stoodley does not explicitly disclose loading the executable object module. It is inherent in Stoodley's computer system in order to execute the compiled code and replace the unresolved reference; wherein the loading includes replacing the unresolved reference with a reference to a system module ("replacing the unresolved code in the compiled code with a call instruction for call the snippet", Col 3, line 54-55, snippet is a system module), and wherein neither the compiling nor the loading including determining whether the unresolved reference refers to a defined external symbol.

c. executing the executable object module ("on execution", Col 2, line 28); wherein the executing includes, calling the system module for loading the separately compiled object module ("call to the corresponding snippet and in which the snippet includes instructions to pass the resolution data to helper code", Col 2, line 53-55); and

d. executing the separately compiled object module ("runtime helper", Col 2, line 15, snippet calls/loads the runtime helper into the execution unit to resolve unresolved reference).

Claim 39: Stoodley discloses the machine-readable medium as in claim 36 above, but does not explicitly disclose the system module is a loader unit. It is inherent in Stoodley's computer system ("snippet includes instructions to pass the resolution data to the runtime helper" Col 2, line 54-55; when the snippet passes resolution data to the helper code, it's actually performing the loading data to the helper code).

Claim 46: Stoodley discloses an apparatus comprising:

- a. although, Stoodley does not explicitly disclose loading a first set of instructions into an execution unit. It is inherent in Stoodley's computer system in order to execute the compiled code emitted by the compiler and replace the unresolved reference; wherein the first set of instructions includes an unresolved reference to a second set of instructions ("compiled code including a set of target instructions...each target instruction including an unresolved reference", Col 2, line 18-21, Stoodley does not explicitly disclose an unresolved reference to a second set of instructions. It is inherent in Stoodley's approach in order to fulfill the purpose of his invention), wherein the loading includes replacing the unresolved reference with an address of a third set of instructions ("replacing the unresolved code in the compiled code with a call instruction for calling the snippet", Col 3, line 54-55, snippet is a third set of instruction);
- b. executing instructions of the first set ("on execution" Col 2, line 28);
- c. executing instructions of the third set ("at runtime" Col 3, line 28-30); wherein executing instructions of the third set includes loading the second set of instructions into the execution unit ("snippet including instructions and data to identify the unresolved

code/reference in the compiled code and to call and provide data to the runtime helper” Col 3, line 48-50, runtime helper is a second module and gets loaded into the execution unit when snippet calls it); and

d. executing instructions of the second set (“runtime helper” Col 2, line 15, snippet calls/loads the runtime helper into the execution unit to resolve unresolved reference).

Claim 47: Stoodley discloses the apparatus as in claim 46 above, and further discloses the first set of instructions includes an executable object module (“compiled code” Col 2, line 18).

Claim 49: Stoodley discloses the apparatus as in claim 46 above, and further discloses the second set of instructions includes a separately compiled object module (“an associated runtime helper being defined for resolving references at runtime” Col 2, line 1-2, the runtime helper is a second module and separately from the compiled code).

Claim 50: Stoodley discloses the apparatus as in claim 46 above, but does not explicitly disclose the third set of instructions includes a loader unit. It is inherent in Stoodley’s computer system (“snippet includes instructions to pass the resolution data to the runtime helper” Col 2, line 54-55; when the snippet passes resolution data to the helper code, it’s actually performing the loading data to the helper code).

Claim 51: although, Stoodley discloses the apparatus as in claim 46 above, but does not explicitly disclose the loading does not include determining whether the unresolved reference refers to a defined external symbol. It is inherent in Stoodley's approach because his computer system identifies the unresolved reference and resolves them at runtime ("identify the unresolved reference and to permit the reference to be resolved by instructions in runtime helper" Col 5, line 5-7).

Claim Rejections - 35 USC § 103

10. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

11. Claims 3, 11, 13, 19, 23, 28, 32, 40, 42, and 48 are rejected under 35 U.S.C. 103(a) as being unpatentable over Stoodley (US 6,813,764), in view of "Apple Developer Connection" Apple Computer Inc. 2001.

Claim 3: Stoodley discloses the method as in claim 1 above, but does not explicitly disclose the executable object module is in the Mach-O object format. However, Mach O file format is well known in the art ("Apple Developer Connection", 2001). It would have been obvious to one having an ordinary skill in the art at the time the invention was made to recognize that Mach-O is a file format for executable object

Art Unit: 2193

code and use it for storing executable object code. The use of Mach-O file format is obvious because one having an ordinary skill in the art would have been motivated to use Mach-O file format to provide both intermediate and final storage of machine code and data. It was designed as a flexible replacement for the BSD a.out format to be used by the compiler.

Claims 11, 13, 19, 23, 28, 32, 40, 42, and 48 depend on the independent claims 7, 12, 17, 21, 26, 30, 36, and 41 respectively, recite the limitation as recited in claim 3, therefore, have been addressed in connection with the rejection of claim 3.

12. Claim 9 is rejected under 35 U.S.C. 103(a) as being unpatentable over Stoodley (US 6,813,764).

Claim 9: Stoodley discloses the method as in claim 7 above, but does not explicitly disclose the loader unit is a dyld loader. However, the applicant recites in the specification page 8, paragraph 1 "dyld, which is a dynamic loader available from Apple Computer, Inc. it would have been obvious to one having an ordinary skill in the art at the time the invention was made to use dyld loader in Stoodley's computer system for the loader unit for loading compiled object code because it is available. Therefore, one of the skills in the art would have been motivated to use dyld loader in Stoodley's computer system since it's available and ready to use. This can save time when developing the code.

13. Claims 10, 20, 18, 27, 29, and 38 are rejected under 35 U.S.C. 103(a) as being unpatentable over Stoodley (US 6,813,764), in view of Wilson et al (US 6,701,515).

Claim 10: Stoodley discloses the method as in claim 7 above, but does not explicitly disclose the source code module include instructions of a dialect of the C programming language. Wilson discloses an analogous method using C programming language. It would have been obvious to one having an ordinary skill in the art at the time the invention was made to combine Wilson's approach in Stoodley's computer system. The combination is obvious because one having a ordinary skill in the art would have been motivated to use C programming language in Stoodley's approach to allow programmers to describe their application in a form that is easy for them to precisely describe (Col 3, line 65-67) and can typically be used with little or no modification on many different types of processors (Col 4, line 4-5).

Claims 18, 27, and 38 depend on the independent claims 17, 26, and 36 respectively, recite the limitation as recited in claim 10 above, therefore, have been addressed in connection with the rejection of claim 10.

Claim 20: Stoodley discloses the apparatus as in claim 17, but does not explicitly disclose the compiler unit compiles Objective C programming language instructions. However, Wilson discloses an analogous apparatus using a compiler to compile Object

Art Unit: 2193

C programming language instructions ("a compiler, as is well known to those versed in the art, converts (compiles) user applications written in High level programming languages such as C" see Wilson Col 3, line 62-64). It would have been obvious to one having an ordinary skill in the art at the time the invention was made to modify the compiler unit in Stoodley's system in order to compile other analogous programming languages such as C. Therefore, one having an ordinary skill in the art would have been motivated to modify Stoodley's compiler unit to be able to compile multiple analogous programming languages.

Claim 29 depends on the independent claim 26, recites the limitation as recited in claim 20 above, therefore, has been addressed in connection with the rejection of claim 20.

14. Claims 12, 14-16, 41, 43-45 are rejected under 35 U.S.C. 103(a) as being unpatentable over Stoodley (US 6,813,764), in view of Sexton et al (US 6,434,685).

Claim 12: Stoodley discloses a method comprises creating an executable object module ("compiled code" Col 2, line 17) that includes symbolic references to addresses in ones of a set of one or more separately compiled object modules ("compiled code including a set of target instructions... each target instruction including an unresolved reference" Col 2, line 17-20); wherein the executable object module includes resolved internal data-to-code offsets (Col 5, line 36-41); replacing the symbolic references with

Art Unit: 2193

addresses to a loader subroutine ("replacing the unresolved code in the compiled code with a call instruction for call the snippet", Col 3, line 54-55, snippet is consider as a loader subroutine); executing the executable object module ("target instruction to be executed" Col 4, line 9), wherein executing includes, executing the loader subroutine to load one of the separately compiled object modules ("on execution the trigger instruction triggers the execution of snippet instructions (defined instructions) to pass data to the runtime helper for resolution of the unresolved reference" Col 2, line 28-30); and executing the one of the separately compiled object modules ("one execution" Col 2, line 18, executing snippet instructions and runtime helper), but does not explicitly disclose the executable object module includes a page-aligned code segment and a page-aligned data segment. Sexton discloses a method for paged memory management system within a runtime environment that solves the page-aligned problem. It would have been obvious to one having an ordinary skill in the art at the time the invention was made to combine Sexton's method with Stoodley' approach to improve the performance. The combination is obvious because one of the skills in the art would have been motivated to recognize that page alignment enables the address of a page to be efficiently calculated from a machine pointer to an object that is allocated in a paged memory. Therefore, saving memory and improving the performance of page associated memory management operations (see Sexton Col 12, line 1-8).

Claim 14: Stoodley and Sexton disclose the method as in claim 12, but do not explicitly disclose the loader subroutine is included in a dynamic loader, and wherein the

Art Unit: 2193

dynamic loader is dyld. However, the applicant recites the availability of dyld, which is dynamic loader from Apple Computer, Inc. in the specification (" dyld, which is a dynamic loader available from Apple Computer, Inc.", page 8, paragraph 1). It would have been obvious to one having an ordinary skill in the art at the time the invention was made to recognize that loader subroutine must include in a dynamic loader and the availability of dyld, dynamic loader from Apple to use in Stoodley's system. Therefore, one having an ordinary skill in the art would have been motivated to use dyld loader in Stoodley's system since it is available and ready to use. This can save time when developing the code.

Claim 15: Stoodley and Sexton disclose the method as in claim 12, but do not explicitly disclose the unresolved reference is a reference is a function call to a function include in one of the separately compiled object modules of the set. However, the applicant commits a prior art in the specification ("unresolved references (e.g., external function calls, references to external variables, etc." page 1, paragraph 2). It would have been obvious to one having an ordinary skill in the art at the time the invention was made to include this feature in Stoodley's approach to clarify the definition of an unresolved reference. Therefore, one having an ordinary skill in the art would have been motivated to state in Stoodley specification to clearly define an unresolved reference is external function calls.

Claim 16: Stoodley and Sexton disclose the method as in claim 12, but do not explicitly disclose the unresolved reference is a reference to a variable defined within one of the separately compiled object modules of the set. However, the applicant commits a prior art in the specification ("unresolved references (e.g., external function calls, references to external variables, etc." page 1, paragraph 2). It would have been obvious to one having an ordinary skill in the art at the time the invention was made to include this feature in Stoodley's approach to clarify the definition of an unresolved reference. Therefore, one having an ordinary skill in the art would have been motivated to state in Stoodley specification to clearly define an unresolved reference is reference to external variables.

15. Claims 13 and 42 are rejected under 35 U.S.C. 103(a) as being unpatentable over Stoodley (US 6,813,764) in view of Sexton et al (US 6,434,685) as applied to claim 12 above, and further view of "Apple Developer Connection" Apple Computer Inc. 2001.

Claim 13: Stoodley and Sexton disclose the method as in claim 12 above, but do not explicitly disclose the executable object module is in the Mach-O object format. However, "Apple Developer Connection" Apple Computer, discloses Mach O file format for storing executable code. It would have been obvious to one having an ordinary skill in the art at the time the invention was made to recognize that Mach-O is a file format for executable object code and use it for storing executable object code. The use of Mach-O file format is obvious because one having an ordinary skill in the art would have

been motivated to use Mach-O file format to provide both intermediate and final storage of machine code and data. It was designed as a flexible replacement for the BSD a.out format to be used by the compiler.

Claims 41-45 are machine-readable medium claims recite limitations as recited in method claim 12-16, therefore, has been addressed in connection with the rejection of claim 12-16.

16. Claim 37 rejected under 35 U.S.C. 103(a) as being unpatentable over Stoodley (US 6,813,764), in view of Chauvel (US 2004/0260911 A1).

Claim 37: Stoodley discloses the machine-readable medium as in claim 36 above, but does not explicitly disclose the determination the address includes looking up the address in a master symbol table. Chauvel discloses an analogous system using symbol table for looking up the unresolved reference (Col 3, paragraph 27). It would have been obvious to one having an ordinary skill in the art at the time the invention was made to modify Stoodley's computer system to include the symbol table as disclose by Chauvel. The modification would have been obvious because one of the ordinary skill in the art would have been motivated to use symbol table to resolve the unresolved reference.

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Phillip H. Nguyen whose telephone number is (571) 270-1070. The examiner can normally be reached on Monday - Friday 10:00 AM - 3:00 PM EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kakali Chaki can be reached on (571) 272-3719. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

PN
10/15/06

Kakali Chaki
Supervisory Patent Examiner



KAKALI CHAKI
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100